Aula – 3
Introdução a Linguagem C

Disciplina: CCO016 - Fundamentos de Programação

Prof: Phyllipe Lima
phyllipe@unifei.edu.br

Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação

Agenda

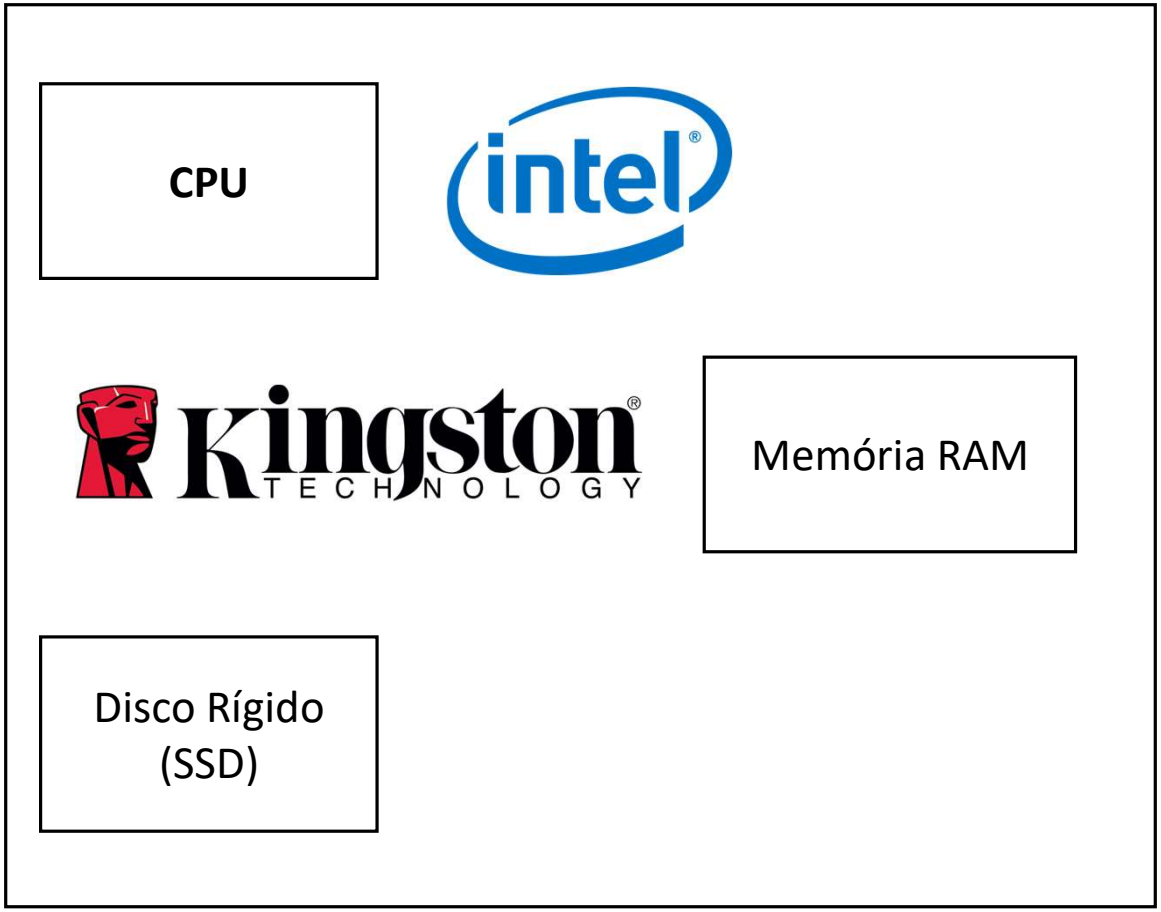
- Computador
- Linguagem C
- Compilação
- Meu Primeiro Programa

Computador

O que é um computador?

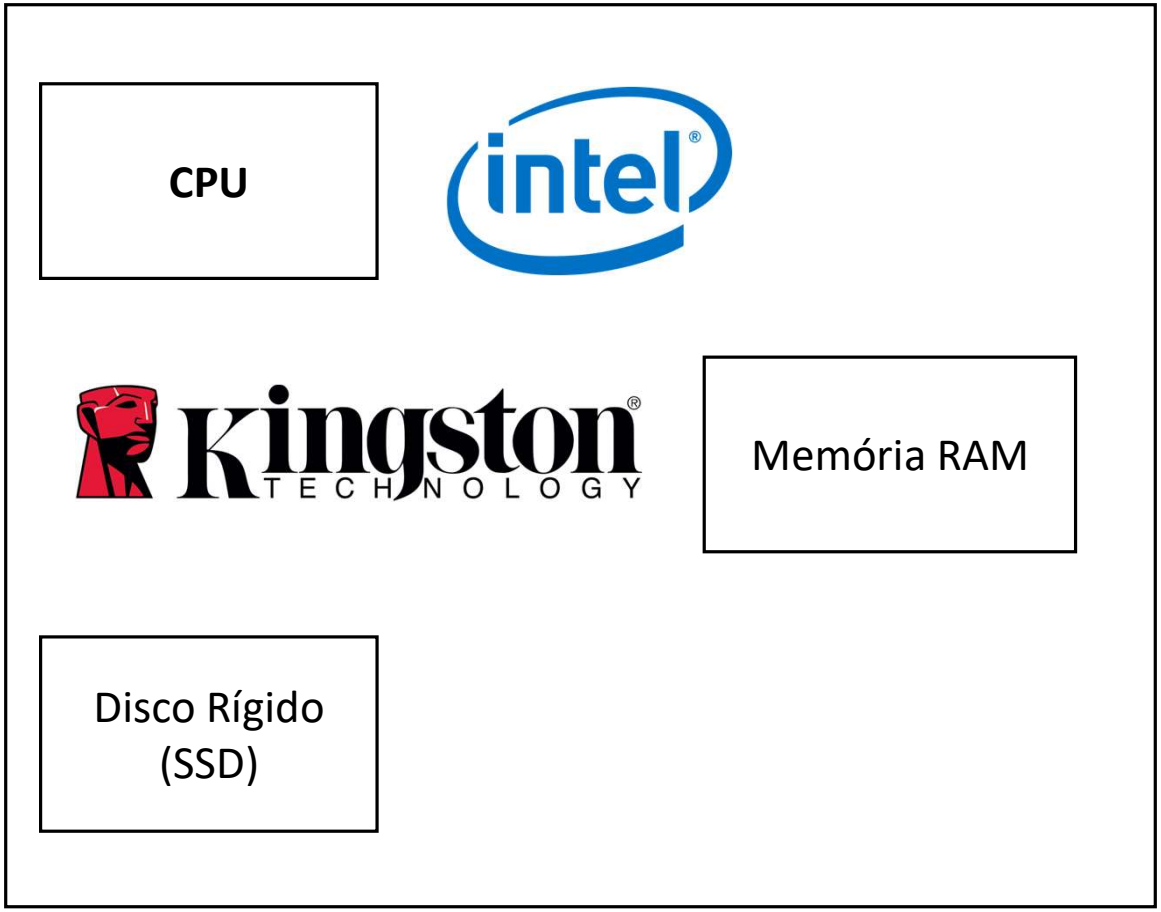
- “Um conjunto de componentes eletrônicos capaz de executar variados tipos de algoritmos e tratamento de informações. Um computador pode possuir inúmeros atributos, dentre eles armazenamento e processamento de dados...”

Computador



- CPU
  - *Central Processing Unit*
  - Unidade de Processamento Central
  - “Cérebro”. Esse componente que executa as instruções

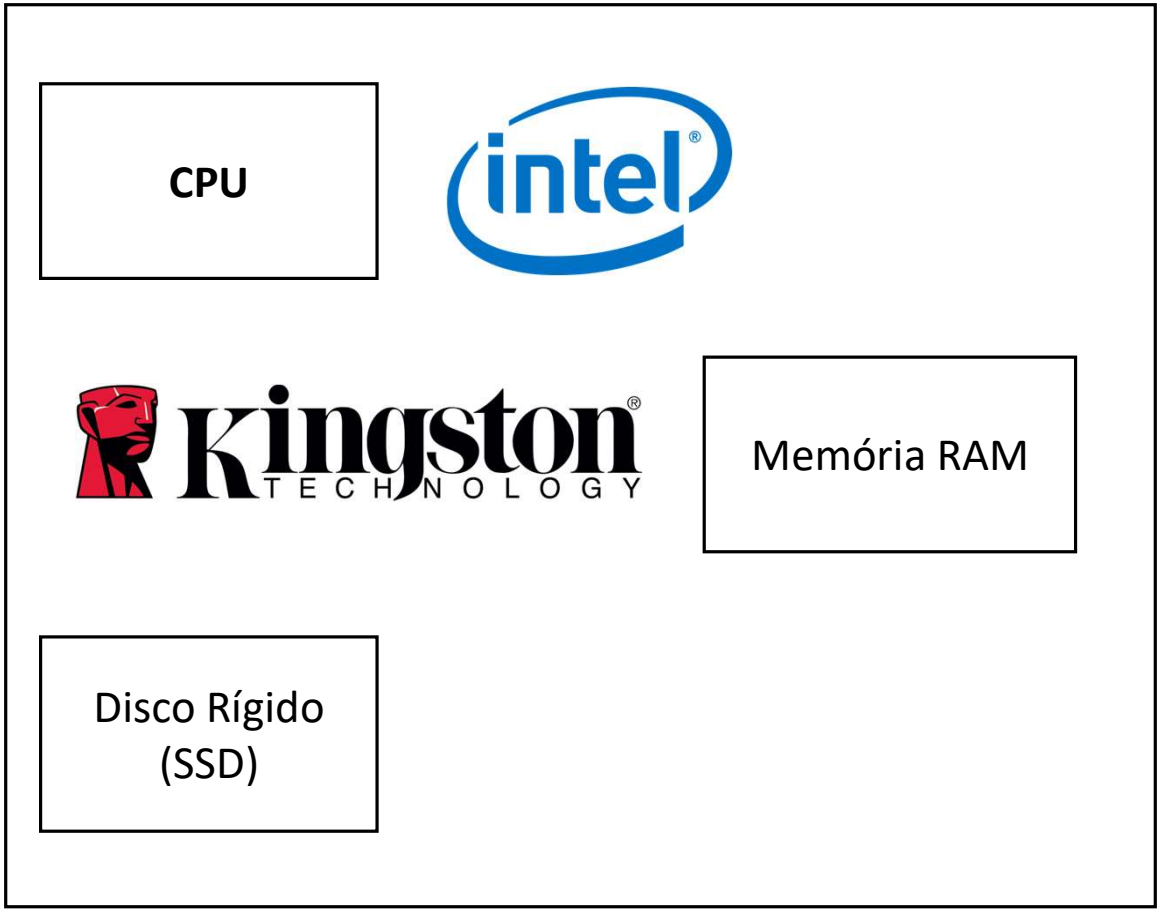
Computador



- CPU
  - *Central Processing Unit*
  - Unidade de Processamento Central
  - “Cérebro”. Esse componente que executa as instruções

- Disco Rígido
  - Armazenamento de dados estático.
  - Grande Capacidade
  - Lento

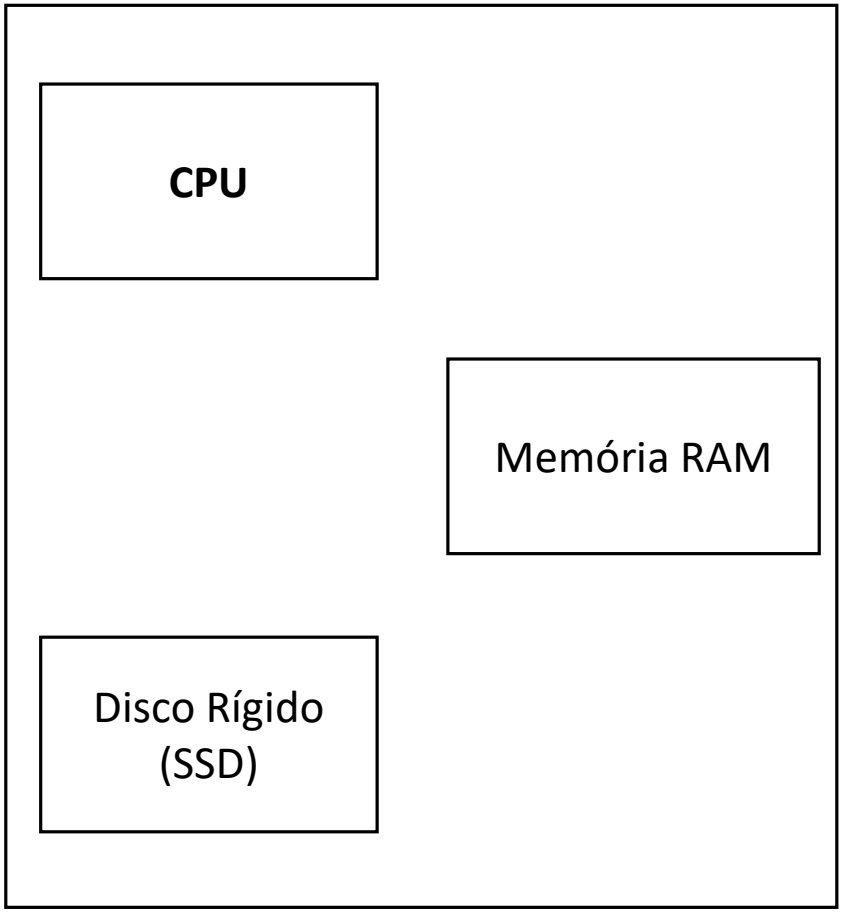
Computador



- CPU
  - *Central Processing Unit*
  - Unidade de Processamento Central
  - “Cérebro”. Esse componente que executa as instruções
- Disco Rígido
  - Armazenamento de dados estático.
  - Grande Capacidade
  - Lento
- Memória Ram
  - Armazenamento de dados de programas em execução
  - Rápida

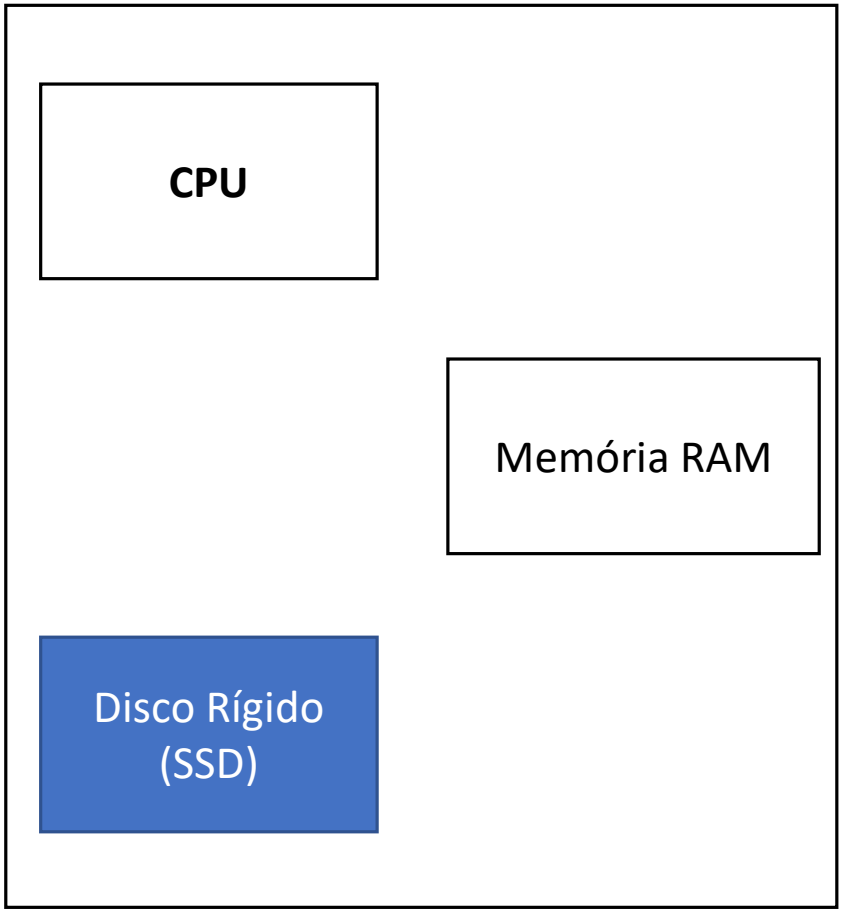
Execução de um Programa

- Inicialmente o programa está “parado” no disco rígido



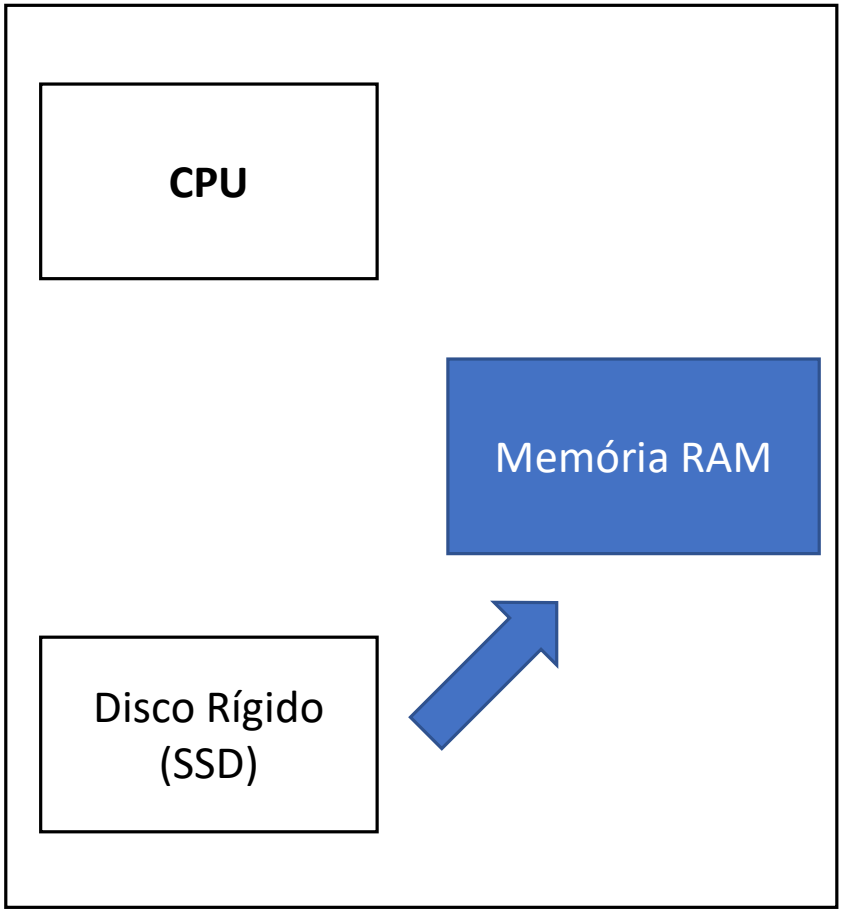
Execução de Programa

- Inicialmente o programa está “parado” no disco rígido



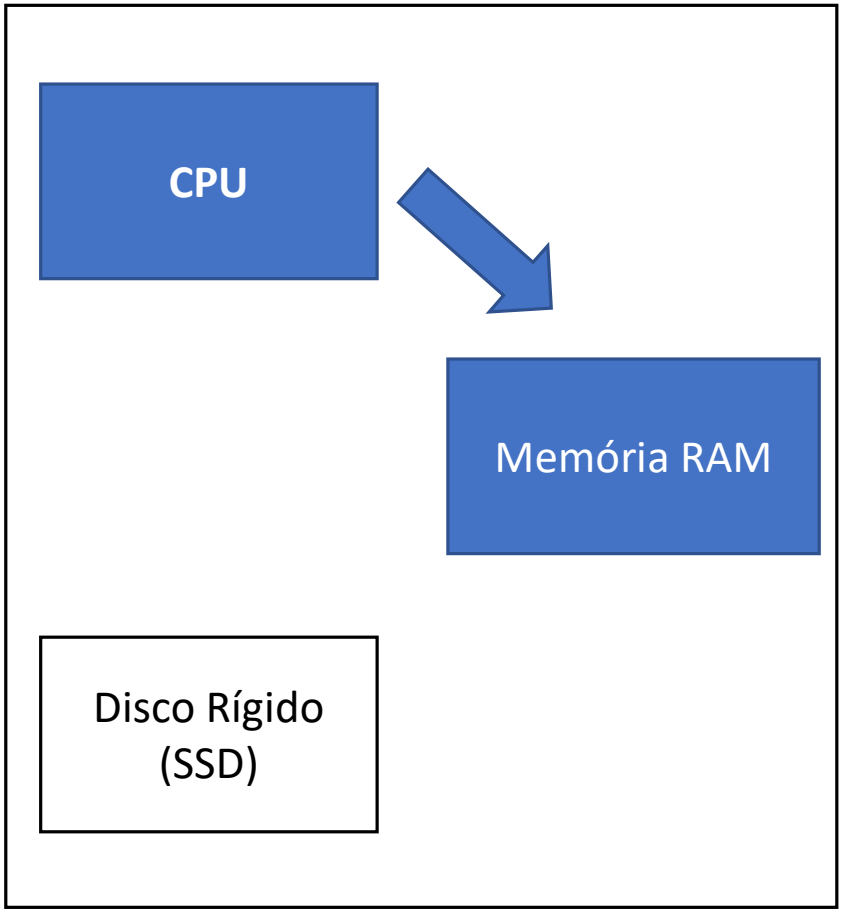
Execução de Programa

- Inicialmente o programa está “parado” no disco rígido.
- Ao “clicar” duas vezes, ele é carregado na memória RAM



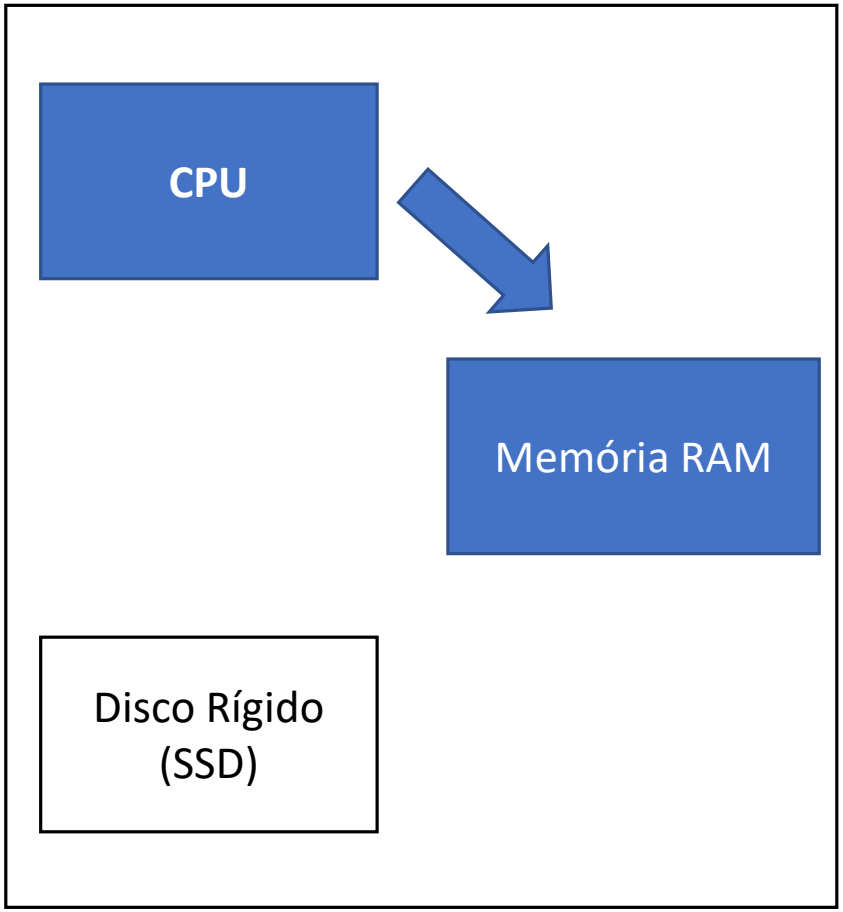
Execução de Programa

- Com o programa na memória, a CPU busca as instruções nela.
- E assim inicia a execução



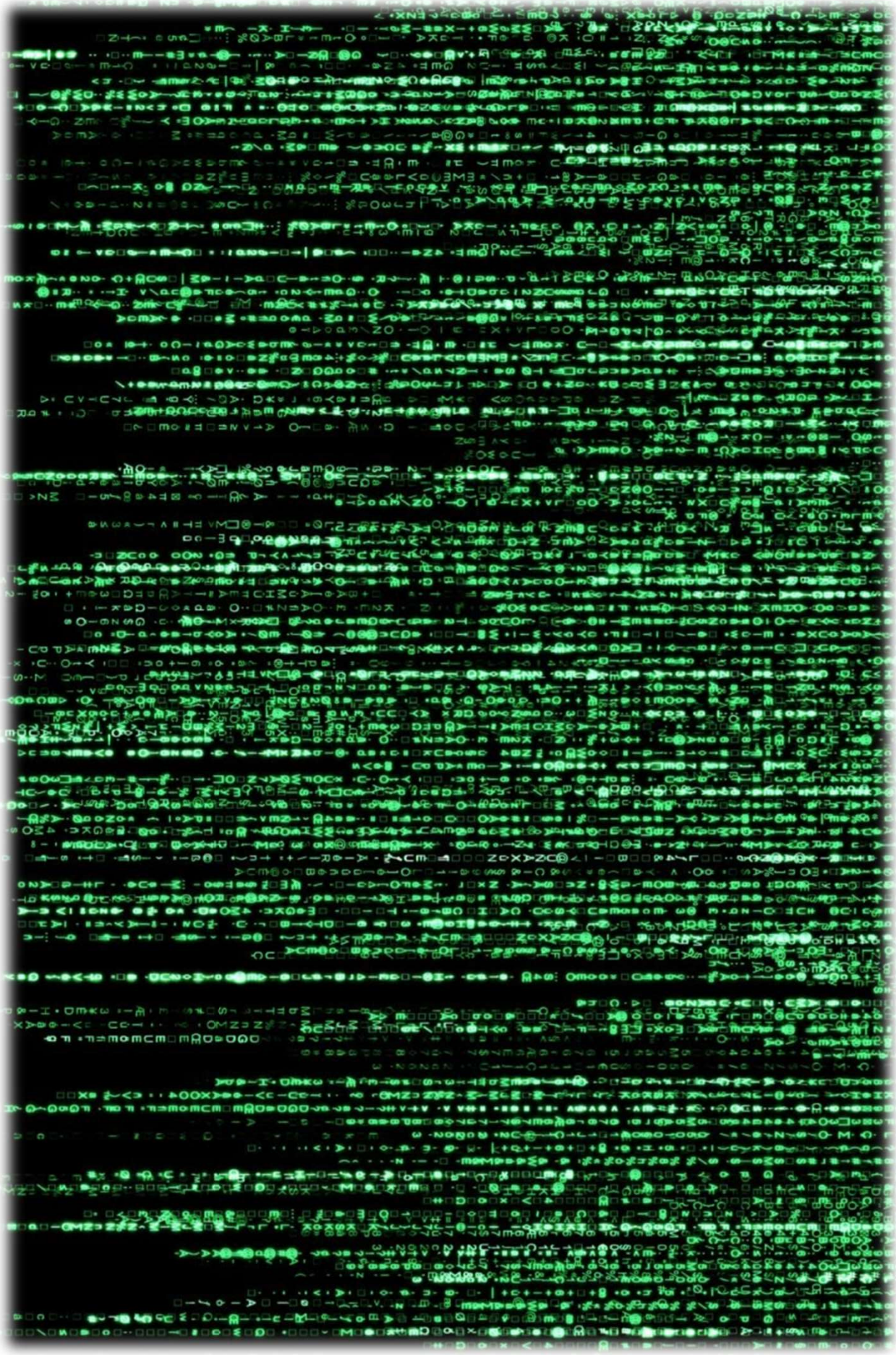
Execução de Programa

- Esse processo permanece até o programa ser encerrado.



Como são escritas as instruções?

- O computador (CPU) entende apenas uma linguagem.
- A linguagem de máquina.



Linguagem de Programação

- É inviável escrevermos programas em linguagem de máquina.
- Por isso criamos as linguagens de programação

Linguagem C

Linguagem de Programação - C

- Umas das primeiras linguagens de programação.
- Desenvolvida para produzir programas fonte mais compacto e organizado.
- Primeira aparição 1972

Linguagem de Programação - C

Código C

```
1. #include <stdio.h>
2. int main()
3. {
4.     printf("Hello!");
5.     return 0;
6. }
```

Código Assembly 8086

```
1.      org 100h
2.      mov dx, offset texto
3.      mov ah, 9
4.      int 21h
5.      ret
6.      texto db "Hello!"
```

Linguagem de Programação - C

Código C

```c
1. #include <stdio.h>
2. int main()
3. {
4.     int x;
5.     scanf("%d", &x);
6.     return 0;
7. }
```

Assembly

```
1.          org 100h
2.          mov cx, 0h
3. entrada:
4.          mov ah, 1
5.          int 21h
6.          cmp al, 13
7.          jz fim
8.          and al, 0fh
9.          mul cx, 0ah
10.         add cl, al
11.         jmp entrada
12.fim:
13.         ret
```

Unix

- A linguagem C foi utilizada para reescrever o sistema operacional Unix.
- Se tornou um dos sistemas mais populares.
- Popularizado por Dennis Ritchie e Ken Thompson.

Unix

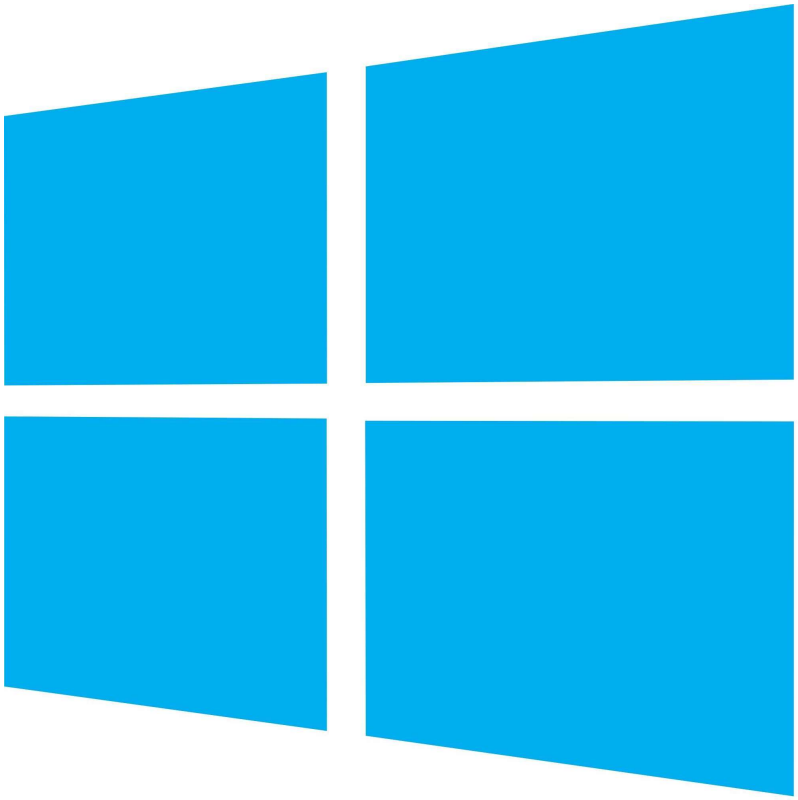
- Sistema Operacional Desenvolvimento Originalmente no *Bell Labs* pela *at&t* da década de 1960.

Unix-Like

- Dada sua popularidade, passou a ser o sistema operacional favorito e foi adotado como parte de outros sistemas, dando origem ao *Unix-like*

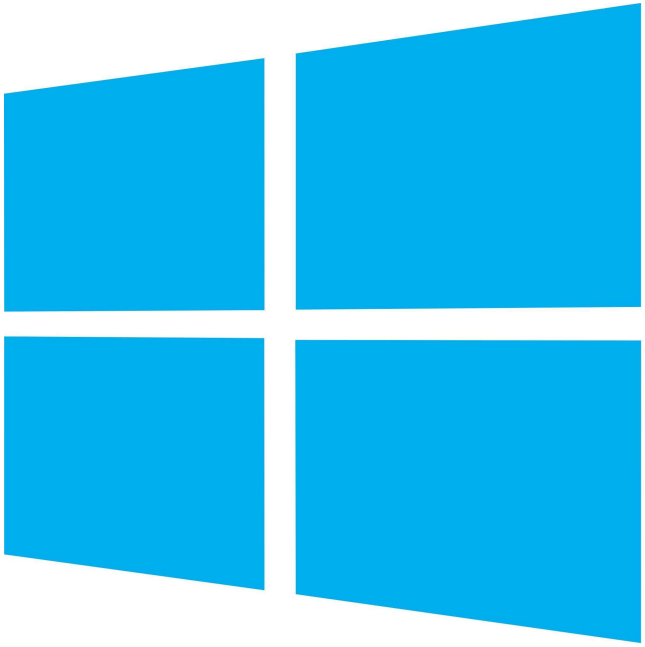
Unix-Like

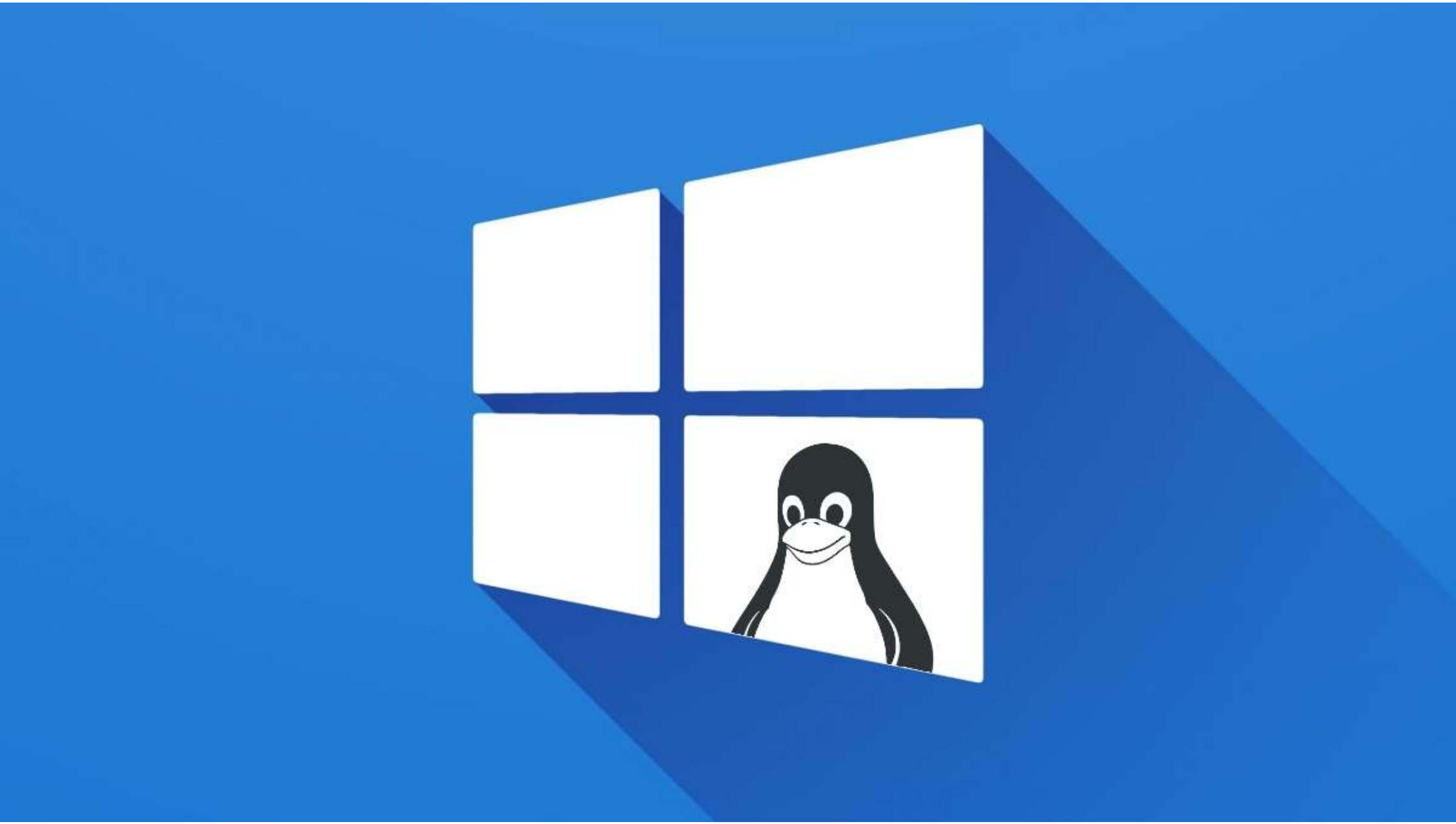
- E o Windows? ☹



Unix-Like

- WSL (Windows Subsystem Linux) para **nos salvar!**





phillima@DESKTOP-CM6Q9KD: /mnt/c/WINDOWS/system32

```
elcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

 System information as of Thu Aug 25 10:08:03 -03 2022

 System load:  0.0                Processes:             8
 Usage of /:   1.3% of 250.98GB   Users logged in:       0
 Memory usage: 0%                 IPv4 address for eth0: 192.168.28.234
 Swap usage:   0%


 updates can be applied immediately.



his message is shown once a day. To disable it please create the
home/phillima/.hushlogin file.
hillima@DESKTOP-CM6Q9KD:/mnt/c/WINDOWS/system32$ LINUX DENTRU DU UINDOUS UHUUUUUUUUU
```

Linguagem Amplamente Utilizada até os dias atuais

Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	^	Python	15.42%	+3.56%
2	1	v	C	14.59%	+2.03%
3	3		Java	12.40%	+1.96%
4	4		C++	10.17%	+2.81%
5	5		C#	5.59%	+0.45%

Desenvolvendo e Executando Programas em C

Fluxo de Execução

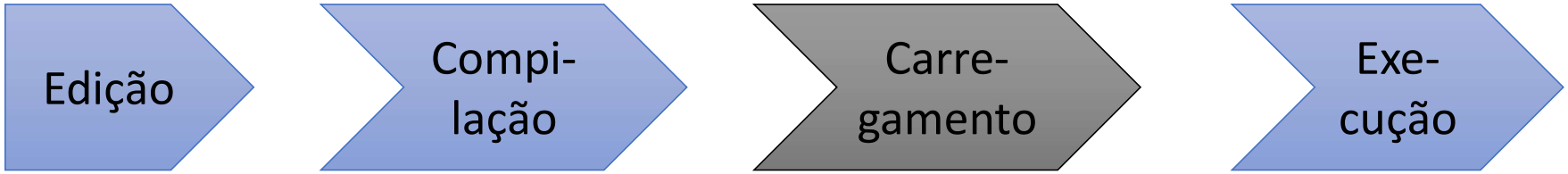
Fluxo de Execução

- **Edição**:
- Consiste na edição do código do programa (humanos entendem)
- É realizada em software de edição de código fonte:
  - Eclipse, CodeBlocks, CLion
- Ou em editores de texto:
  - VSCode,Notepad++, Vim, emacs, Kate, etc.
- Arquivos tem a extensão `.c`.

Fluxo de Execução

- **Compilação**:
- Fase em que ocorre a tradução do programa para linguagem de máquina.
- O compilador verifica o código em busca de erros de sintaxe.
- Se houverem erros, o programa não é traduzido:
  - Erros de compilação.
- Caso contrário, é gerado um executável (a CPU entende)

Fluxo de Execução



- **Carregamento**:
- Esta fase é realizada por um software do sistema operacional conhecido como `loader` (carregador).
- Este software busca o arquivo executável no disco e o transfere para a memória principal (RAM).
- Adicionalmente, também são carregadas bibliotecas utilizadas no programa.

Fluxo de Execução

- **Execução**:
- Após estar carregado na memória, o programa é executado.
- Esta fase ocorre sob o controle da CPU.

Ambiente VSCode

VSCode – Visual Studio Code

- Nas aulas, utilizaremos o VSCode para nossos experimentos.

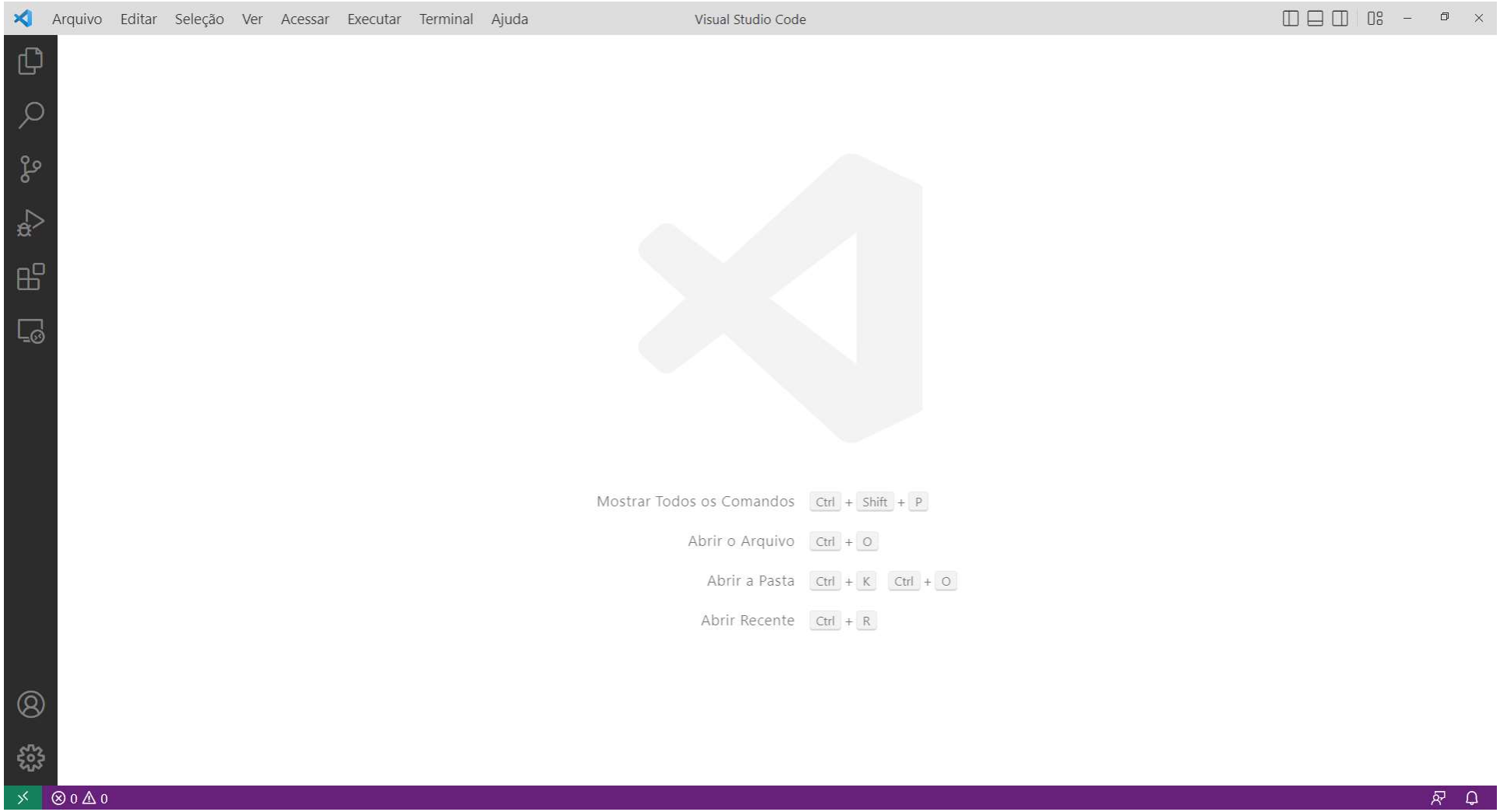
VSCode – Visual Studio Code

- Próximos slides mostram o VSCode em uma máquina Windows 10 com WSL instalado.
- Será comentado também alternativa de ambientes *online*

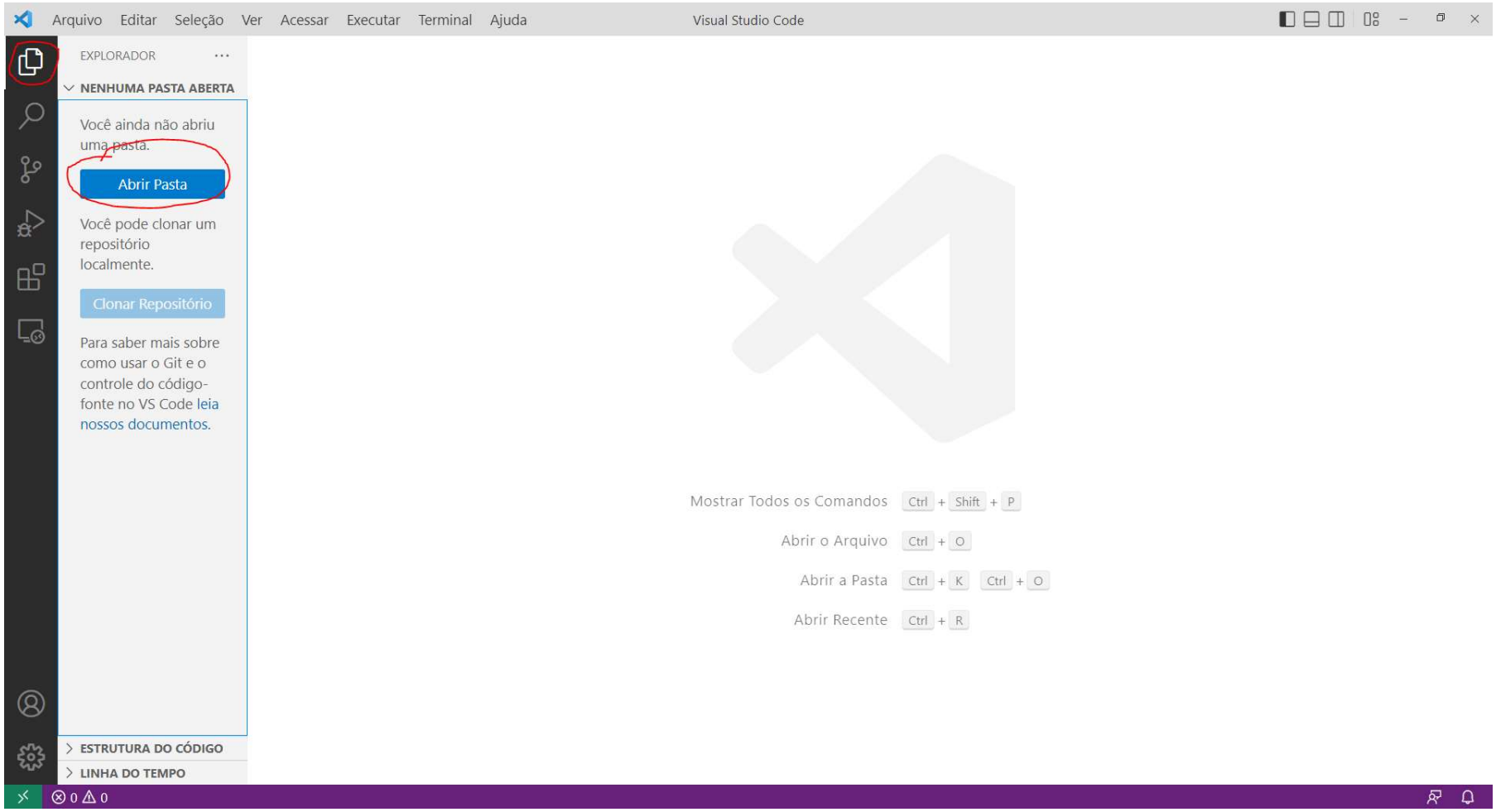
VSCode – Visual Studio Code

- Próximos slides mostram o VSCode em uma máquina Windows 10 com WSL instalado.
- Será comentado também alternativa de ambientes *online*

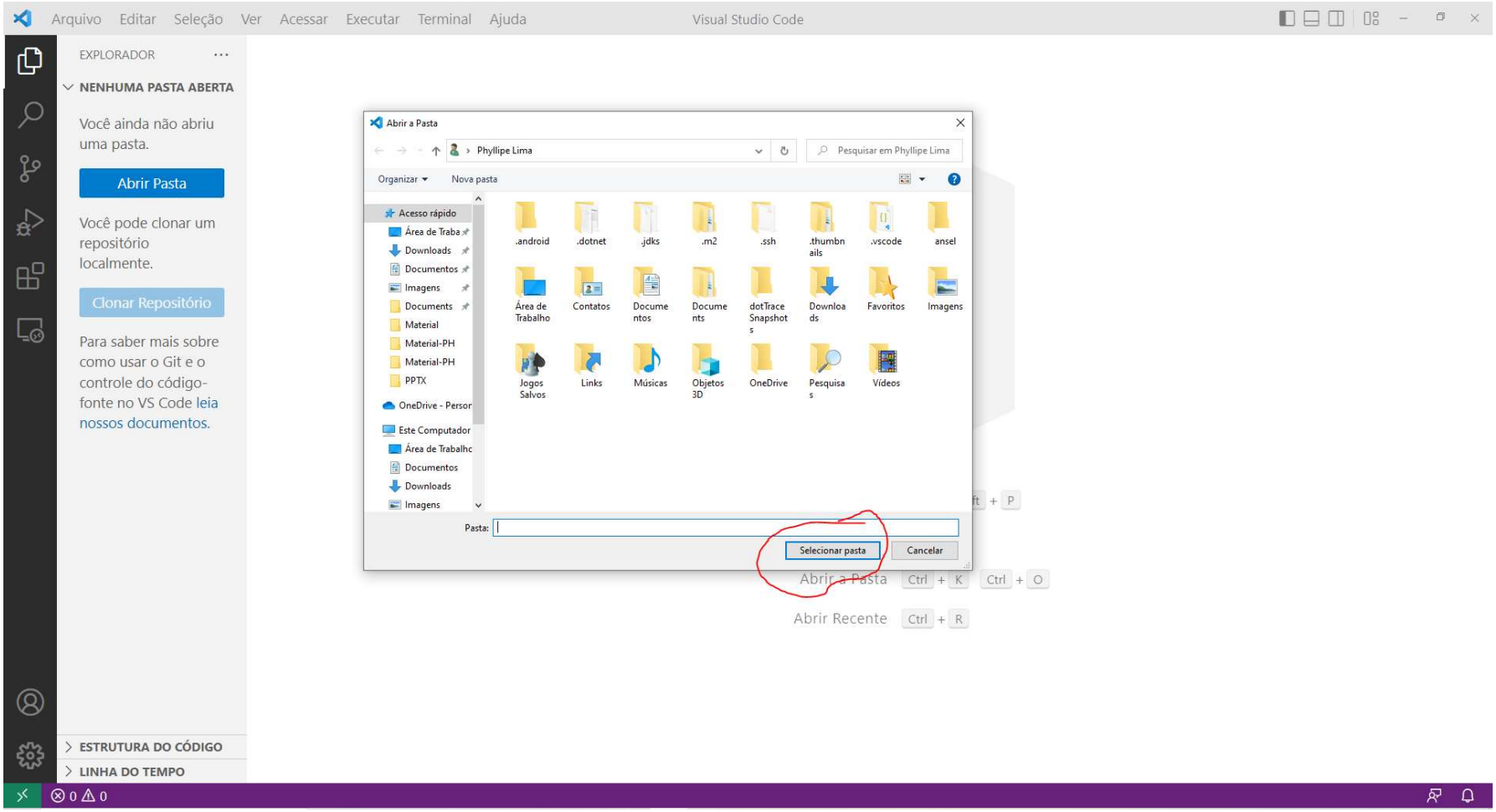
Após abrir o VSCode



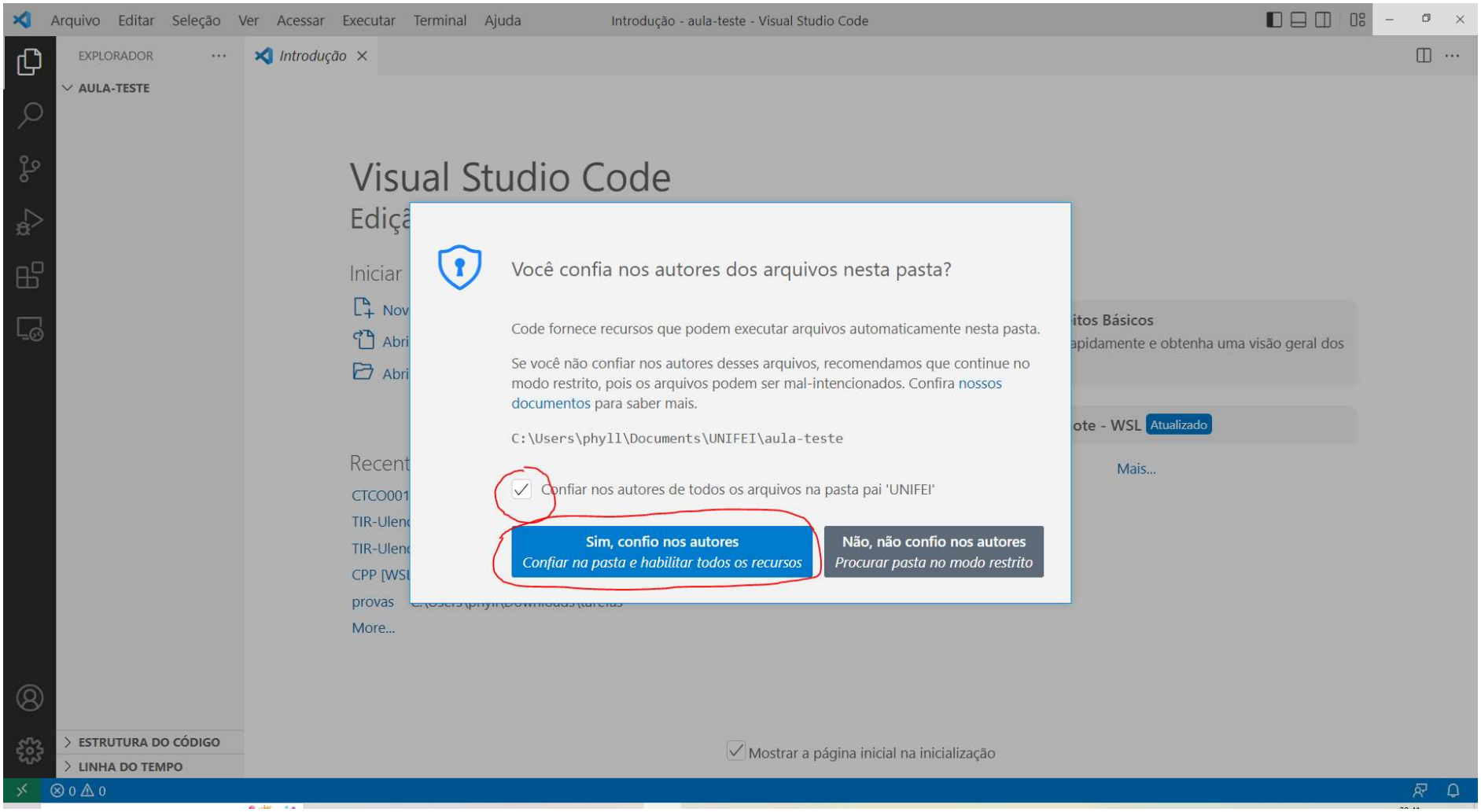
Criando uma pasta para seus arquivos



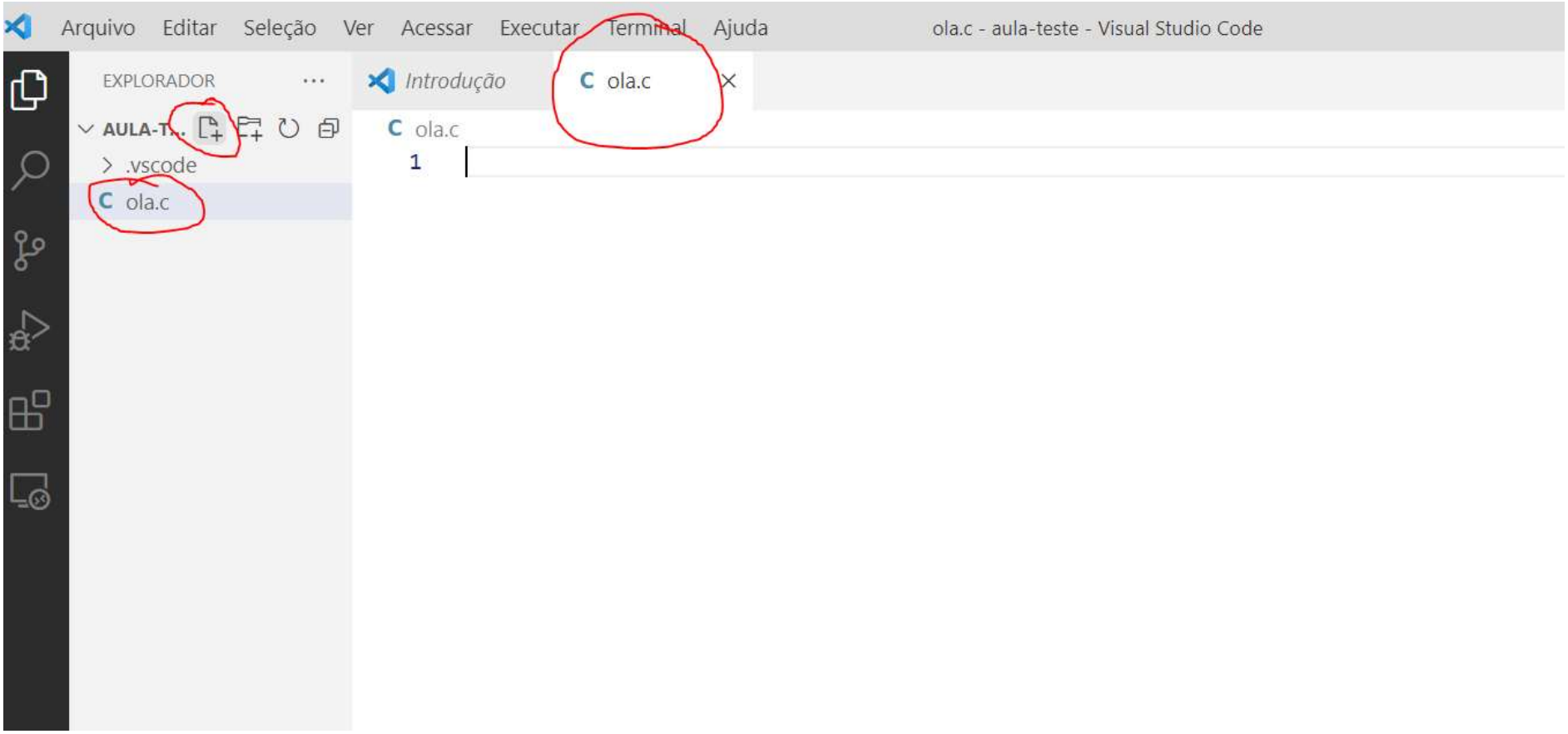
Escolha um local e clique “Selecionar Pasta”



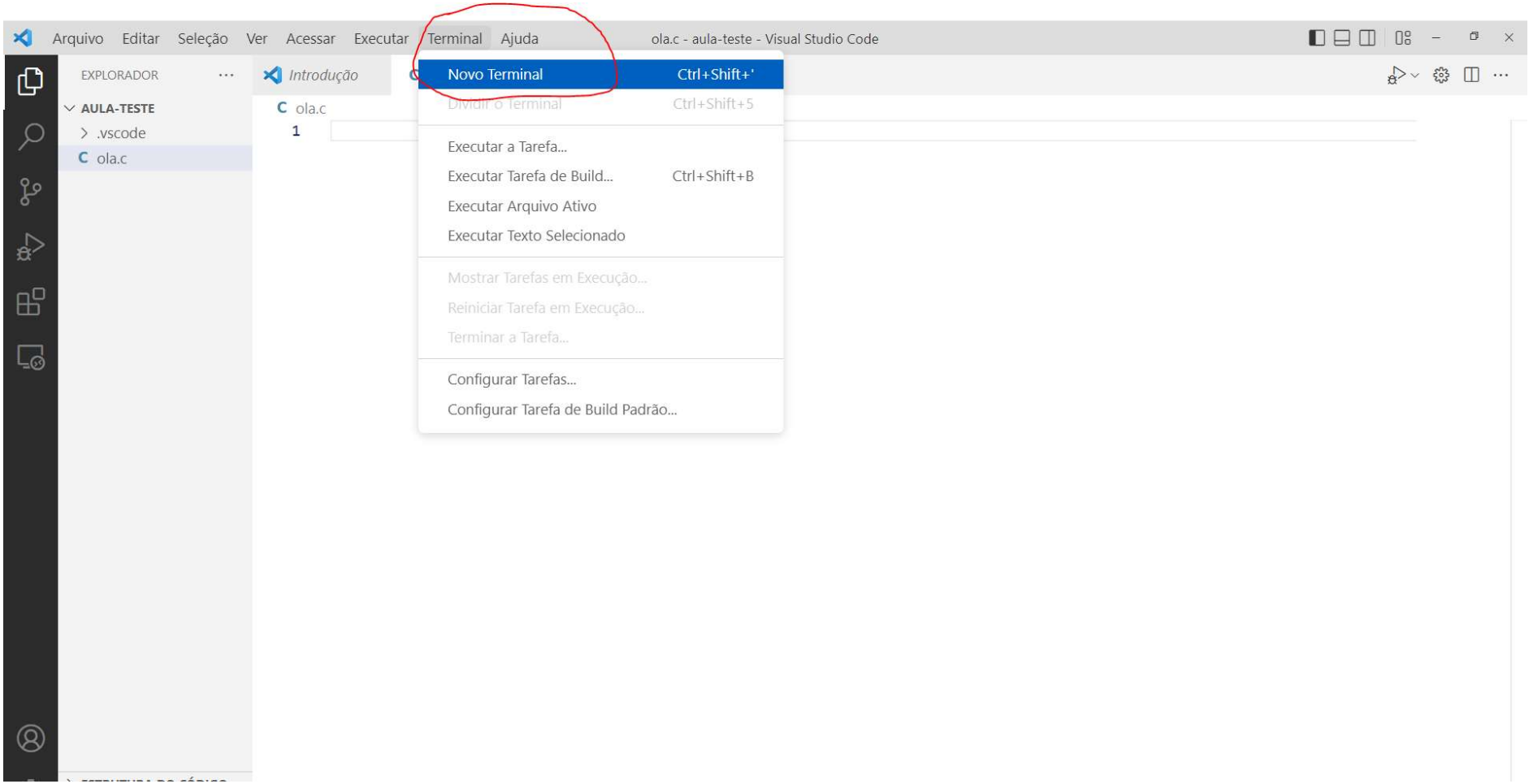
Marque que “sim” você confia nos autores



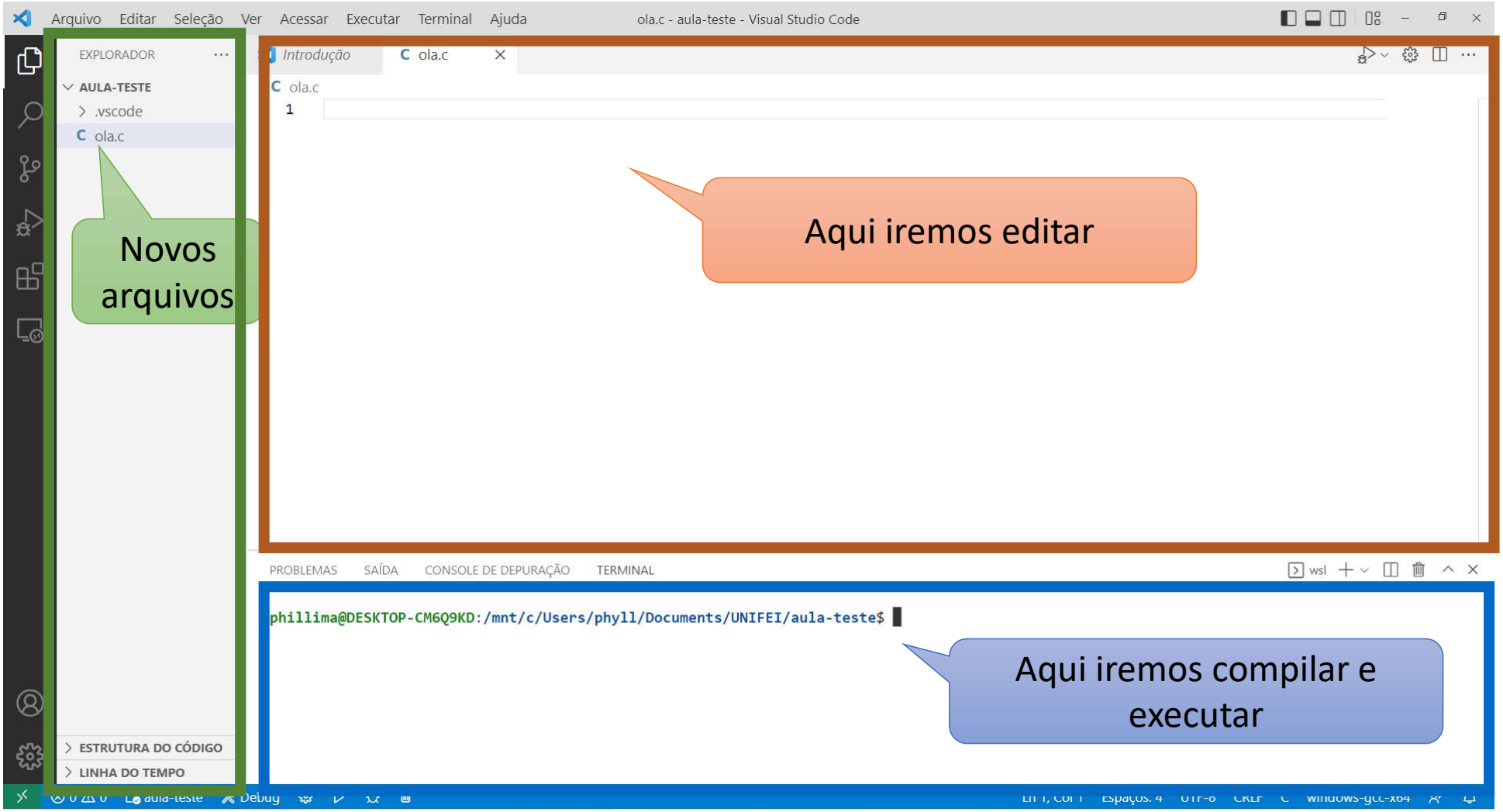
Crie seu primeiro arquivo “ola.c”



Abra o terminal (requer WSL instalado)



Ambiente pronto ☺



Ambiente Online

Online C Compiler.

- Entre no link: https://www.onlinegdb.com/online_c_compiler
- Seja feliz.
- Suficiente para nosso curso. Mas não reflete ambiente real para programas corporativos.

Clique em “Run”

Aqui iremos compilar e executar

OnlineGDB beta
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

86.8K

Run | Debug | Stop | Share | Save | { } Beautify

Language C

main.c

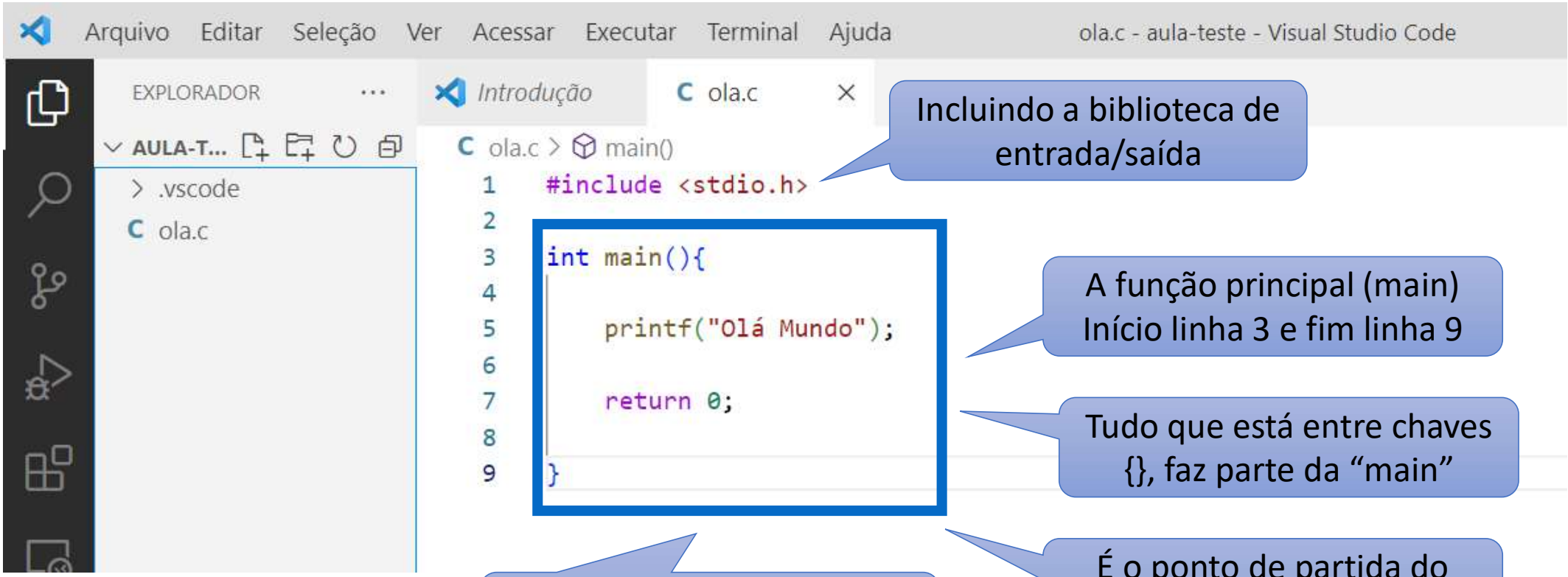
```
/
                    Online C Compiler.
            Code, Compile, Run and Debug C program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************

#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```

Aqui iremos editar

Meu Primeiro Programa

Meu Primeiro Programa

Meu Primeiro Programa



Incluindo a biblioteca de entrada/saída

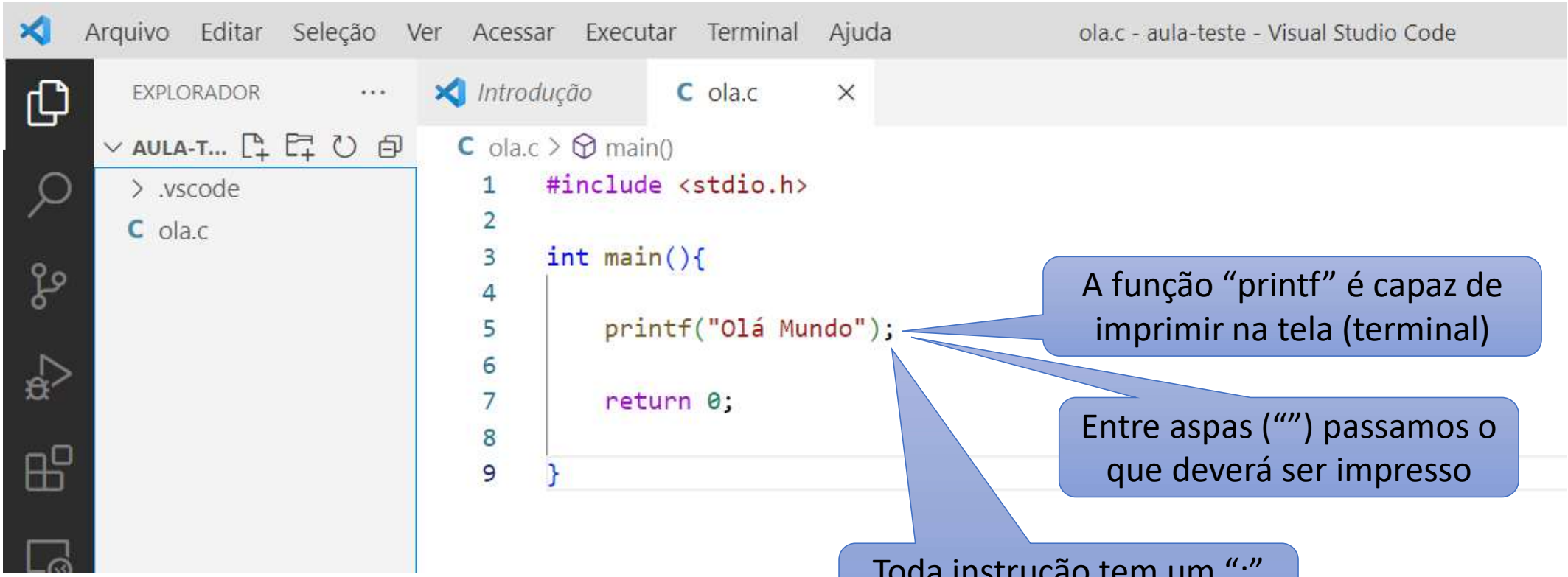
A função principal (main)
Início linha 3 e fim linha 9

Tudo que está entre chaves {}, faz parte da “main”

É o ponto de partida do programa

Tudo entre {}, faz parte do *corpo* da função

Meu Primeiro Programa

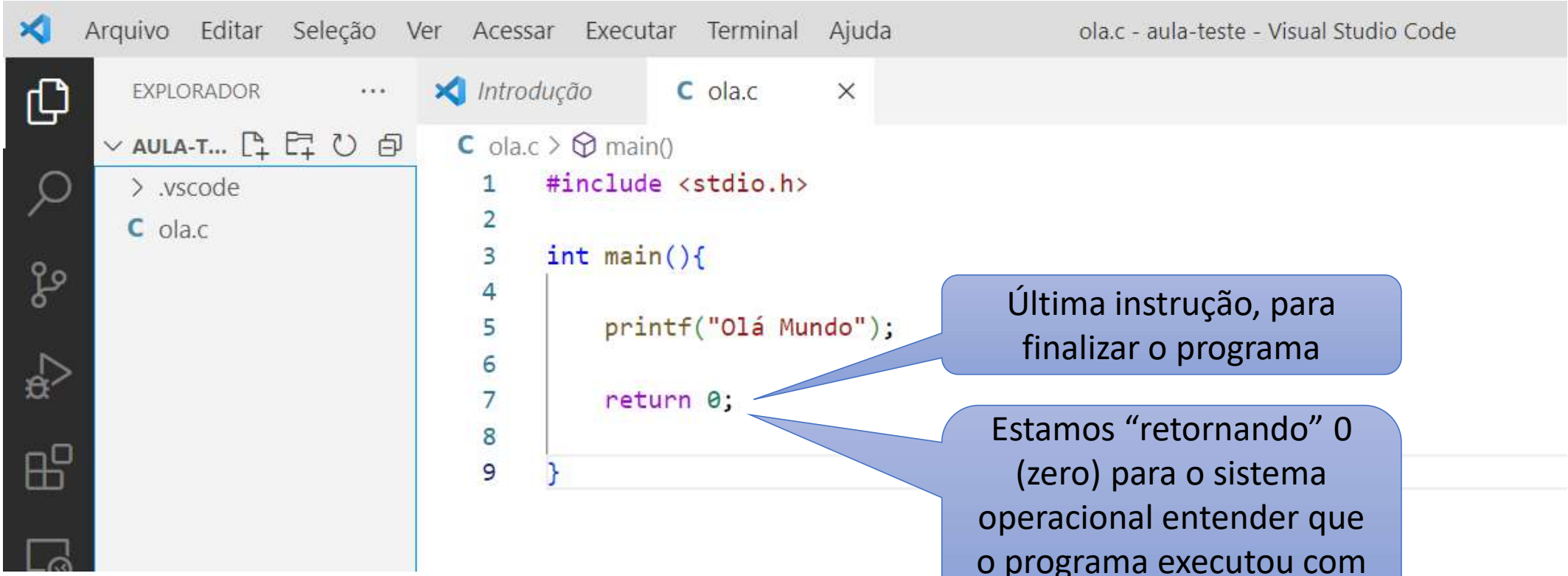


A função “printf” é capaz de imprimir na tela (terminal)

Entre aspas (“”) passamos o que deverá ser impresso

Toda instrução tem um “;” (ponto e vírgula) no final

Meu Primeiro Programa



Última instrução, para finalizar o programa

Estamos “retornando” 0 (zero) para o sistema operacional entender que o programa executou com sucesso

Meu Primeiro Programa

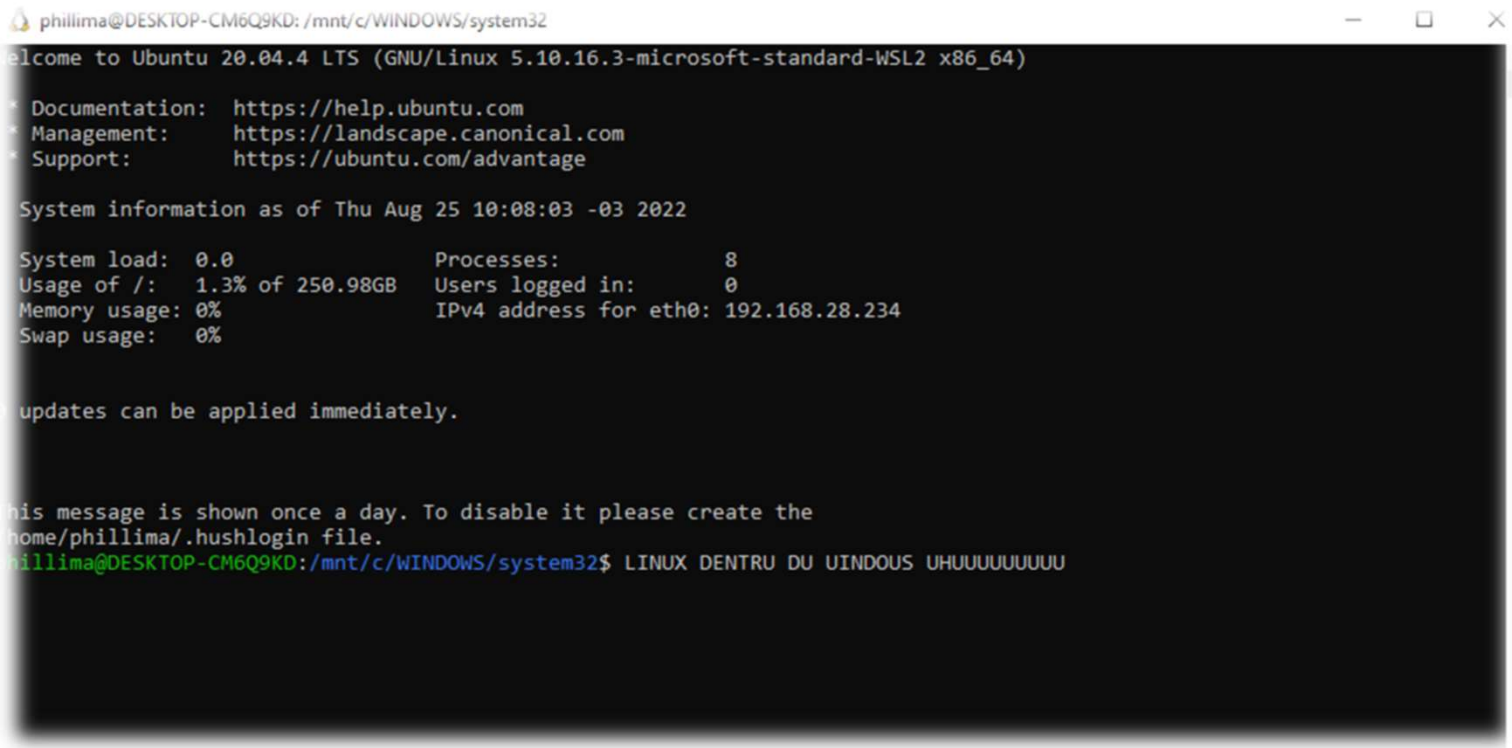
- Este código está escrito em Linguagem C. Como podemos *traduzir* para que a CPU consiga executar?
- Através da compilação!

Meu Primeiro Programa

- Este código está escrito em Linguagem C. Como podemos *traduzir* para que a CPU consiga executar?
- Através da compilação!

Meu Primeiro Programa

- #partiu #terminal



Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda ola.c - aula-teste - Visual Studio Code

EXPLORADOR

AULA-TESTE
- .vscode
- ola.c
- prog

Introdução | ola.c

ola.c > main()

```c
#include <stdio.h>

int main(){

    printf("Olá Mundo\n");

    return 0;

}
```

O comando “ls” irá listar os arquivos e diretórios

Certifique que houve uma saída com o nome do seu arquivo. Nesse exemplo “ola.c”

PROBLEMAS SAÍDA DE DEPURAÇÃO TERMINAL

```
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ls
ola.c
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ gcc ola.c -o prog
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ./prog
Olá Mundo
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$
```

ESTRUTURA DO CÓDIGO

LINHA DO TEMPO

0 0 aula-teste Debug Ln 5, Col 24 Espaços: 4 UTF-8 CRLF C windows-gcc-x64

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda ola.c - aula-teste - Visual Studio Code

EXPLORADOR

AULA-TESTE
- .vscode
- ola.c
- prog

Introdução | ola.c

ola.c > main()

```c
#include <stdio.h>

int main(){

    printf("Olá Mu

    return 0;

}
```

Faça a compilação através de:
gcc nomeDoArquivo.c –o nomeDoExecutável

Nesse exemplo "nomeDoArquivo.c" é "ola.c" e o "nomeDoExecutável" é "prog"

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ls
ola.c
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ gcc ola.c -o prog
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ./prog
Olá Mundo
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$
```

ESTRUTURA DO CÓDIGO
LINHA DO TEMPO

0 0 aula-teste Debug Ln 5, Col 24 Espaços: 4 UTF-8 CRLF C windows-gcc-x64

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda ola.c - aula-teste - Visual Studio Code

EXPLORADOR

AULA-TESTE
- .vscode
- ola.c
- prog

Introdução | ola.c

ola.c > main()

```c
#include <stdio.h>

int main(){

    printf("Olá Mundo\n");

    return 0;

}
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ls
ola.c
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ gcc ola.c -o prog
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ./prog
Olá Mundo
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$
```

Para executar faça “./prog”, onde prog é o nome do executável

ESTRUTURA DO CÓDIGO

LINHA DO TEMPO

0 0 aula-teste Debug Ln 5, Col 24 Espaços: 4 UTF-8 CRLF C windows-gcc-x64

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda ola.c - aula-teste - Visual Studio Code

EXPLORADOR

AULA-TESTE
- .vscode
- ola.c
- prog

Introdução | ola.c

ola.c > main()

```c
#include <stdio.h>

int

}
```

Resultado:
Olá Mundo

PROBLEMAS | CONSOLE DE DEPURAÇÃO | TERMINAL

```
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ls
ola.c
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ gcc ola.c -o prog
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$ ./prog
Olá Mundo
phillima@DESKTOP-CM6Q9KD:/mnt/c/Users/phyll/Documents/UNIFEI/aula-teste$
```

ESTRUTURA DO CÓDIGO
LINHA DO TEMPO

0 0 aula-teste Debug Ln 5, Col 24 Espaços: 4 UTF-8 CRLF C windows-gcc-x64

Comentários

- Podemos escrever informações para documentar o código através de comentários.
- O compilador irá ignorar os comentários. Tendo valor apenas para humanos.
- Podemos comentar linhas individuais usando // (duas barras)
- Podemos comentar grupos de linhas usando /* (na primeira linha) e */ na última linha.

Introdução | ola.c

ola.c > main()

```c
#include <stdio.h>

int main(){
    //Comentário individual
    printf("Olá Mundo\n");

    return 0;

    /*
    grupo
    de
    linhas
    comentadas
    */

}
```

Comentário Individual

Grupo de linhas comentadas

Aula – 3
Introdução a Linguagem C

Disciplina: CCO016 - Fundamentos de Programação

Prof: Phyllipe Lima
phyllipe@unifei.edu.br

Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação